AD-A274 747

Prof. Kurt VanLehn & Dr. Joel Martin
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260

January 7, 1994

DTIC
S ELECTE
JAN 13 1994
A

Dr. Susan Chipman
Scientific Officer Code 1142CS
Office of Naval Research
800 N. Quincy St.
Arlington, VA 22217-5000

Dear Susan,

This letter is the ninth quarterly progress report for grant N00014-91-J-1532, entitled "On-Line Assessment of Expertise." It covers the period October 1, 1993 to December 31, 1993.

In this period, we have continued the development of the assessment methods for the problem classification, qualitative problem solving, and difficulty estimation activities, whose design was discussed in the preceding quarterly progress report. In addition, we have prepared two journal papers, one of which has been submitted to Communications of the ACM for publication in a special issue devoted to applications of uncertain reasoning.

For one of these papers, we have developed a detailed example of Olae assessing problem solving. This example has also been converted so that it may be presented on the World Wide Web via Internet. It will be available when LRDC has completed its WWW server. This example is enclosed.

Best regards,

Kurt VanLehn
Assoc. Prof. of Computer Science
LRDC Senior Scientist
&
Joel Martin
LRDC Research Scientist

cc: ONR Resident Representative N66005, OSU Research Center
Director, Naval Research Laboratory, Code 2627
Defense Technical Information Center
University of Pittsburgh Contracts Office

Table 1: The rule library for the simple physics domain.

```
mag-cos-rule:    proj(S,x) = mag(S) * cos(incline(S)).
cos-rule:        proj(S,x) = cos(incline(S)).
sum-all-rule:    net-force(Body,Axis) = sum-over(foreach:F in:Set addend:proj(F,Axis))
                         :- find-all-forces(Body, Set).
sum-one-rule:    net-force(Body,Axis) = proj(F1,Axis)
                         :- find-force(Body, F1).
sum-two-rule:    net-force(Body,Axis) = proj(F1,Axis) + proj(F2,Axis)
                         :- find-force(Body, F1),
                            find-force(Body, F2),
                            F1 <> F2.
```

Table 2: A simple Newtonian physics problem of combining multiple forces acting on one physical body.

Two forces act on a point object as follows:
40 N at 45° and 25 N at 60°.
Find the magnitude of the resulting force.

1 An example assessment

As an example, consider the simple domain model shown in Table 1. This domain model has five rules in a shorthand representation for Horn clauses. In this representation, the head of the clause is an equation and the body is a set of conditions. Any quantity in the equation in the head of the rule may be designated as the consequent of the rule at any time during rule use. All other quantities in the equation then become additional antecedents of the clause. In essence, this representation is a shorthand for several Horn clauses. The first rule in Table 1 is equivalent to three Horn clauses, each with a different quantity as the consequent.

In the sample domain model (Table 1), the first two rules concern how to project a vector onto an axis to determine a component of the magnitude. The first rule correctly states that the X component of the vector is the product of the vector's magnitude and the cosine of its angle with the x-axis. The second rule incorrectly omits the influence of the vector's magnitude. The last three rules describe how the same axis components of vectors can be combined along an axis. The third rule correctly states that the resulting magnitude on the axis is the sum over the same axis components from all forces acting on the physical body. The fourth and fifth rules are overly specific and are only correct when there are just one or two forces (respectively) acting on the body.

For the problem shown in Table 2, the first and second steps of the analysis generate a Bayesian net like that shown in Figure 1. The rules are shown in dashed boxes and the given parts of the problem are on the left side of the figure. The rules are labeled with their rule-names. The possible actions are on the right side and represent various interpreted equations that a student could type. Here they are represented with no substitutions. The intermediate nodes represent rule applications (labeled 'RA') and intermediate conclusions.

Suppose that a particular student is new to OLAE and has a default student model (Figure 2) with all rules having a probability of 0.33. The first two rules are mutually exclusive as are the third and fourth, and the third and fifth. When the student is presented with the problem in Table 2, the student model is connected to the problem solution graph producing the Bayesian net in
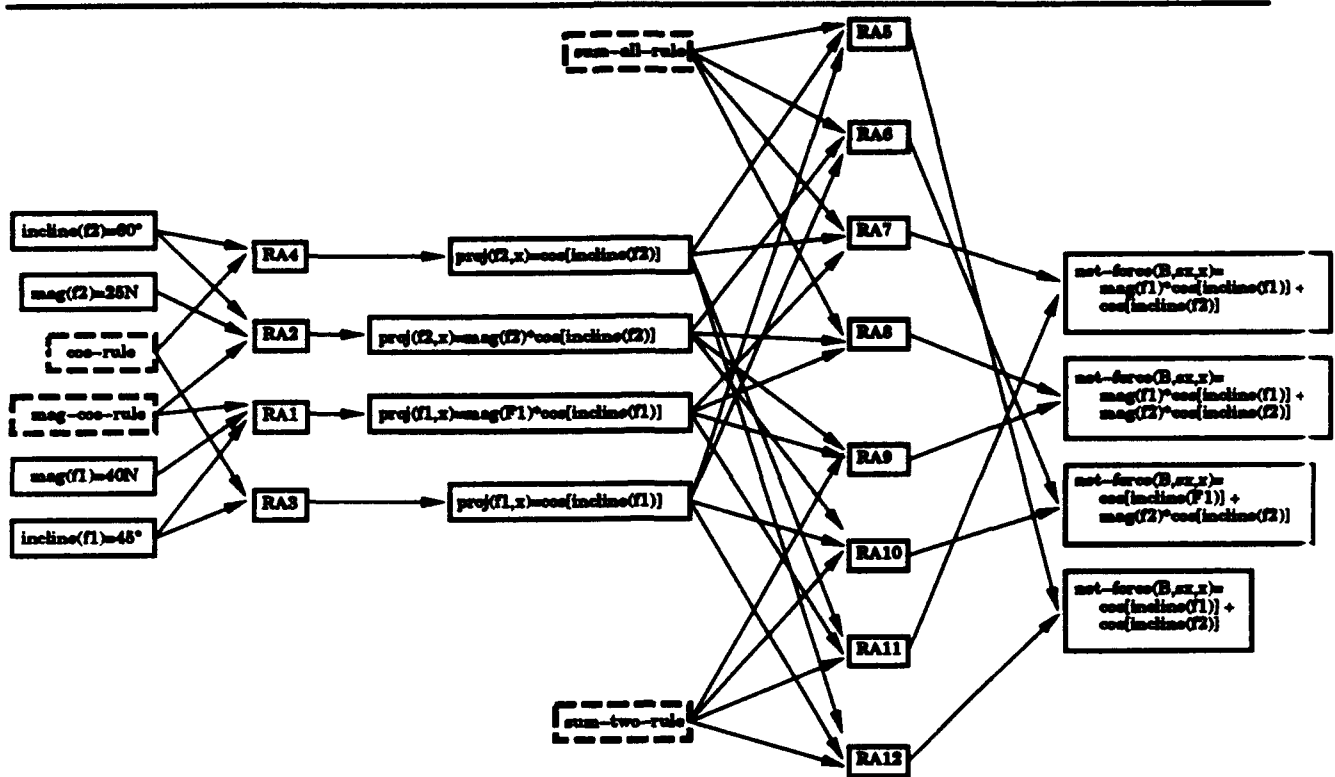
Figure 1: A belief net created for all four rules in Table X in the context of a particular problem.

Figure 3.

Suppose that the student writes the equation, $F = cos(45) + cos(60)$. This equation is put into canonical form. It is used as a key to the database of possible equations and it matches an equation corresponding to the node labeled "$netforce(B, sx, x) = cos[incline(f1)] + cos[incline(f2)]$" (bottom-most node in the last column of Figure 3). A new node is created to represent the action and a deterministic link is created from the equation node to the new Action node. The new equation node is clamped to a probability of 1.0, and this new information is propagated backward to the rules. The rule labeled 'cos-rule' in·Figure·3, has its probability raised from 0.33 to 0.94. In this case, the student is almost certainly using the incorrect rule, 'cos-rule' and not the correct rule 'mag-cos-rule' (Prob = 0.03). The rules 'sum-two-rule' and the 'sum-all-rule' are equally likely with a probability of 0.49, because the observed equation could have been generated by either. Finally, the probability of the rule 'sum-one-rule' decreases from 0.33 to 0.228. After updating the probabilities, the whole network will be carried along as the student model. OLAE can compact the Bayesian nets when they grow too large.

In Figures 2, 3, and 4 the nodes that are highly probable (0.8..1.0) have a thick border, the nodes that are moderately probable (0.5..0.8) have a medium border, and the remaining nodes are low probability.

Suppose a second problem is analyzed and attached to the network (Figure 4). The second problem is a simpler, related problem in which the student only had to compute the x-component of a single force. Again, student actions for this problem are matched (via canonicalization) to the equation Fact nodes, a new Action node is created, that node is clamped to a probability of 1.0 and the net's probabilities are updated.
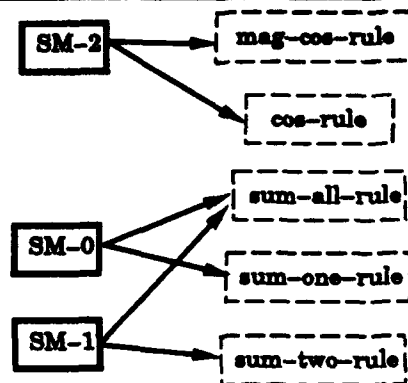
Figure 2: The initial state of the student model for the rules in Table 5. It represents the prior probabilities on the rules, including the fact that the first two rules are mutually exclusive as are the third and fourth, and the third and fifth.

Suppose that the student typed the equation, $F = cos(75)$. Now, the probability that the 'cos-rule' is known has increased from 0.94 to 0.998. The probability that the 'sum-all-rule' is known has also increased from 0.49 to 0.77 because it is consistent with both observations. The probability of the 'sum-two-rule' has decreased to 0.206 and the probability of the 'sum-one-rule' has not changed. Furthermore, the overall Bayesian net now represents the fact that the student either has the 'sum-all-rule' or both the 'sum-one-rule' and the 'sum-two-rule'. If OLAE later sees evidence to discredit 'sum-all-rule', then it immediately has a strong belief in both the other rules. After analyzing the second problem, OLAE uses the whole Bayesian net as the new student model, and the process repeats once for every remaining problem.
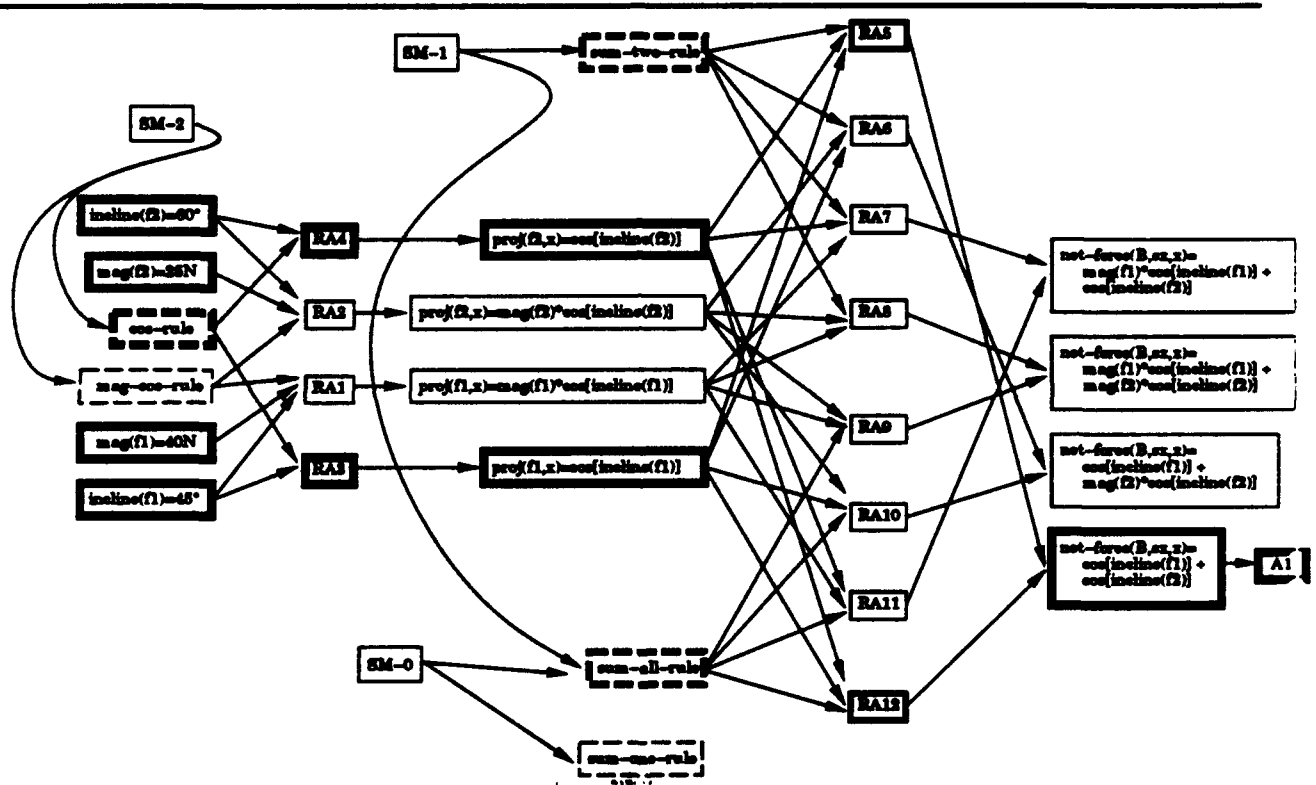
Figure 3: The state of the student model for the rules in Table 1 after the first problem.

Figure 4: The state of the student model for the rules in Table 1 after the second problem.